

\$8.88 Estimated cost File351
\$0.19 TYMNET
\$9.07 Estimated cost this search
\$19.27 Estimated total session cost 0.459 DialUnits

File 347:JAPIO OCT 1976-2001/Apr(UPDATED 010801)

(c) 2001 JPO & JAPIO

*File 347: JAPIO data problems with year 2000 records are now fixed.

Alerts have been run. See HELP NEWS 347 for details.

Set	Items	Description
---	---	-----
?ss	pn=(63073450 or 1140256)	
S1	1	PN=63073450
S2	1	PN=1140256
S3	2	PN=(63073450 OR 1140256)

?t s3/4/all

3/4/1

FN- DIALOG(R)File 347:JAPIO|
CZ- (c) 2001 JPO & JAPIO. All rts. reserv.|
TI- MEMORY PARITY CHECKING METHOD
PN- 01-140256 -J P 1140256 A-
PD- June 01, 1989 (19890601)
AU- HONMA MASAHIRO
PA- FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
AN- 62-298332 -JP 87298332-
AN- 62-298332 -JP 87298332-
AD- November 26, 1987 (19871126)
IC- -4- G06F-012/16; H03M-013/00
CL- 45.2 (INFORMATION PROCESSING -- Memory Units); 42.4 (ELECTRONICS --
Basic Circuits)
SO- Section: P, Section No. 926, Vol. 13, No. 391, Pg. 117, August 30, 1989
(19890830)
AB- PURPOSE: To reduce the number of memories to a half by storing data
requiring parity data not in the bit direction of the memory but in
the direction of a word.

CONSTITUTION: An address generating circuit 2 forms an address for
making access to the data of eight bits and applies to the memory 1.
Thereby, the data outputted after the access time of the memory 1 is
initially latched by a latch circuit 4. This timing is generated by a
latch control circuit 3 by considering the access time. Thereafter,
the circuit 2 generates the address of the parity data corresponding
to the latched data. In this case, the highest bit of the address of
the memory 1 may be simply changed. In such a way, the read parity
data and the previously latched data of the eight bits are inputted
to a parity check circuit 5 to check an error and if there is the
error, an error signal is generated. In such a way, the parity is
checked by the use of the simple control circuit to reduce the number
of the memories to a half.

3/4/2

FN- DIALOG(R)File 347:JAPIO|
CZ- (c) 2001 JPO & JAPIO. All rts. reserv.|
TI- MEMORY WRITING/READING SYSTEM
PN- 63-073450 -JP 63073450 A-
PD- April 04, 1988 (19880404)
AU- HIRASHIMA TAKESHI

PA- HITACHI LTD [000510] (A Japanese Company or Corporation), JP (Japan)
AN- 61-218781 -JP 86218781-
AN- 61-218781 -JP 86218781-
AD- September 17, 1986 (19860917)
IC- -4- G06F-012/16; G06F-011/08
CL- 45.2 (INFORMATION PROCESSING -- Memory Units); 45.1 (INFORMATION
PROCESSING -- Arithmetic Sequence Units)
SO- Section: P, Section No. 745, Vol. 12, No. 300, Pg. 135, August 16, 1988
(19880816)
AB- PURPOSE: To detect errors in a simple constitution by defining a memory
as an original area and a duplicate area, storing the normal data in
the original area and storing the normal data and its address value
in the duplicate area in the form of the check data.

CONSTITUTION: The control information received by a disk controller 3
is inputted to a processor 4 in the controller 3. The processor 4
transmits the normal data serving as the control information and at
the same time produces the check data from said normal data and its
address value as they are or by compressing these data and address
value after arithmetic. Then both normal and check data are stored in
an original area and a duplicate area defined by a program within a
memory circuit 5. Both normal and check data are read out of the
circuit 5 and the presence or absence of address/data errors is
checked.

?logoff

```
12aug01 14:52:11 User116074 Session D4923.4
    $2.07    0.189 DialUnits File347
    $3.00    2 Type(s) in Format  4
    $3.00    2 Types
$5.07  Estimated cost File347
$0.40  TYMNET
$5.47  Estimated cost this search
$24.74 Estimated total session cost  0.648 DialUnits
```

Status: Signed Off. (6 minutes)

⑫ 公開特許公報 (A)

昭63-73450

⑤Int.Cl.⁴G 06 F 12/16
11/08

識別記号

3 1 0
3 1 0

府内整理番号

B-7737-5B
A-7368-5B

⑬公開 昭和63年(1988)4月4日

審査請求 未請求 発明の数 1 (全11頁)

⑭発明の名称 メモリ書き込み・読み出し方式

⑮特願 昭61-218781

⑯出願 昭61(1986)9月17日

⑰発明者 平島 健 神奈川県小田原市国府津2880番地 株式会社日立製作所小田原工場内

⑯出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑯代理人 弁理士 磯村 雅俊

明細書

1. 発明の名称

メモリ書き込み・読み出し方式

2. 特許請求の範囲

1. 記憶領域に重要なデータを書き込む際のメモリ書き込み・読み出し方式において、上記記憶領域上でプログラムにより定義された正領域と副領域を有し、該正領域にはプログラムにより正規のデータを書き込み、また該副領域には、上記正規データと該正規データを書き込んだアドレス値とを、そのままの形あるいは演算して圧縮された形にし、チェックデータとして書き込み、データ読み出し時には、プログラムにより正規データと該正規データに対応するチェックデータとを読み出した後、該チェックデータを基にして、該正規データに対するエラーの有無を調べることを特徴とするメモリ書き込み・読み出し方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、メモリ書き込み・読み出し方式に関するもので、特に電子または回路の不良により発生するエラーを高い確率で検出できるメモリ書き込み／読み出し方式に関するものである。

〔従来の技術〕

ディスク制御装置等においては、装置内にバッファメモリを備えており、このバッファメモリ上には制御装置が制御しているディスク装置の管理情報を記録している。管理情報は極めて重要であるため、このバッファメモリの電子不良やメモリ入出力制御回路の不良により、メモリの記憶内容の変化、つまりデータ化けを起すことは絶対に許されない。このためには、ハードウェアによるチェック回路の増設が必要であるが、実装上の制限からハードウェアの増設には限度がある。

第2図は、従来のメモリおよびその周辺回路のブロック図であり、第3図は第2図のメモリに書き込まれるデータとアドレスとの関係を示す図であり、第4図は従来のメモリおよび改良された周辺回路のブロック図である。

第2図の回路と第4図の回路との相違点は、第2図では回路素子が全て1組ずつ設けられており、アドレス/データのチェック機能はパリティ・チェック回路で行われるのに対して、第4図では等価な回路素子を2個ずつ用いて、一方を正記憶回路とし、他方をアドレス/データのチェックのための副記憶回路としている点である。

以下、第2図と第4図の回路の動作を詳述する。先ず、第2図の回路では、プログラムがデータ書き込み命令を実行すると、ハードウェアはプログラムが指示したアドレス値 A_n と書き込みデータ D_n (およびパリティ P_n) を各々アドレス・レジスタ6と書き込み用レジスタ8に入力する。そして、アドレス・レジスタ6の出力値でアドレスシングしたメモリ10上のアドレスに書き込み用レジスタ8の出力値を、第3図に示すように書き込む。すなわち、アドレス A_0, A_1, \dots, A_n には、それぞれデータ $D_0 P_0, D_1 P_1, \dots, D_n P_n$ が書き込まれる。アドレス・レジスタ6の出力値と書き込み用レジスタ8の出力

ラーの見逃し確率を極力下げるために、第4図の2重化回路が提案された。

第4図では、正記憶回路に属する書き込み用レジスタ15、アドレス・レジスタ14、メモリ16、および読み出し用レジスタ17と、これらに対応する副記憶回路に属する書き込み用レジスタ22、アドレス・レジスタ20、メモリ23、および読み出し用レジスタ24は、各々全く等価な回路であり、常に両方が同じ値を保持している。

メモリ16と23へのデータ書き込み時には、アドレス・レジスタ14と20の出力値をアドレス・コンペア回路19で、また書き込み用レジスタ15と22の出力値を書き込みデータ・コンペア回路18で、それぞれ比較する。

メモリ16と23からのデータ読み出し時には、アドレス・レジスタ14と20の出力値をアドレス・コンペア回路19で、また読み出し用レジスタ17と24の出力値を読み出しデータ・コンペア回路21で、それぞれ比較する。そして、比較結果が不一致であれば、アドレスまたはデータに

値は、各々アドレス・パリティ・チェック回路7とデータ・パリティ・チェック回路9とでチェックされ、エラーの有無がプログラムに知らされる。

次に、プログラムがデータ読み出し命令を実行すると、ハードウェアはプログラムが指示したアドレス値 A_n をアドレス・レジスタ6に入力する。そして、アドレス・レジスタ6の出力値でアドレスシングしたメモリ10上のアドレスからデータ D_n (およびパリティ P_n) を読み出し、読み出し用レジスタ11に入力する。その際に、アドレス・レジスタ6の出力値と読み出し用レジスタ11の出力値は、各々アドレス・パリティ・チェック回路7と読み出しデータ・パリティ・チェック回路12とでチェックされ、エラーの有無がプログラムに通知される。

しかし、第2図の回路では、アドレス値とデータ値に対するチェックとしてパリティ・チェックしか行っていないため、アドレス値またはデータ値に偶数ビットのエラーがあった場合には、エラー検出できない。そこで、アドレス/データ・エ

ラーがあったと判断する。このようにすれば、正副記憶回路の少なくとも一方が正常に動作する限り、アドレス/データ・エラーを検出することができる。

このように、第4図の2重化記憶回路では、第2図の回路に比べると、アドレス/データのエラーに対する検出率は高くなるが、ハードウェア量が多くなるという問題がある。

従来のこの種の方式として、例えば、特開昭57-182256号公報に記載の「外部記憶装置へのデータ記憶方式」では、データ種別ごとにデータをバッファレジスタに一時記憶し、一杯となったならば外部記憶装置に転送するとともに、テーブル記憶部の対応するエリアをクリアすることにより、少ないバッファレジスタで対応できるようしている。また、特開昭57-182863号公報記載の「外部メモリ制御方式」では、外部メモリから計算機の主メモリへの書き込みを、リードアフタライト方式として、通常可変情報部への書き込みを複数箇所で行うことにより、計算機と

外部メモリ間の情報交換時の信頼性を高めている。また、特開昭57-185553号公報記載の「制御装置の外部メモリー制御方式」では、大容量の情報を格納する外部メモリの制御回路において、アドレスカウンタを設置することにより、メモリアドレス指定の効率化を図り、かつ保守の簡単化を図っている。

〔発明が解決しようとする問題点〕

前述したように、第2図の回路では、アドレス経路、またはデータ経路における偶数ビット・エラーを検出できないという問題がある。また、第4図の回路では、2重化回路で構成しているため、2重化回路中の対応する2つ回路が同一の故障とならない限り、アドレス/データ・エラーは必ず検出できるが、2重化するためにハードウェア量が第2図の回路の2倍以上となってしまい、小型化が不可能になるという問題がある。

本発明の目的は、これらの問題を解決し、必要最小限のハードウェア量で、かつ2重化回路と同じ程度のエラー検出が行えるメモリ書き込み・読み

み出し方式を提供することにある。

〔問題点を解決するための手段〕

上記目的を達成するため、本発明のメモリ書き込み・読み出し方式は、プログラムにより定義された正領域と副領域を有し、該正領域にはプログラムにより正規のデータを書き込み、また該副領域には、上記正規データと該正規データを書き込んだアドレス値とを、そのままの形あるいは演算して圧縮された形にし、チェックデータとして書き込み、データ読み出し時には、プログラムにより正規データと該正規データに対応するチェックデータとを読み出した後、該チェックデータを基にして、該正規データに対するエラーの有無を調べることに特徴がある。

〔作用〕

本発明においては、ハードウェア量を2倍にすることなく、プログラムがメモリ上で正エリアと副エリアとを定義しておき、上記正エリアにはプログラムにより正規のデータを書き込み、副エリアには正規のデータおよび正規データを書き込ん

だアドレス値とを、そのままの形で、あるいは加工した形で、これをチェックデータとして書き込む。データ読み出し時には、プログラムが正規データとチェックデータを両方とも読み出し、プログラムにより正規データに対するチェックデータの関係からエラーの有無を調べる。

〔実施例〕

以下、本発明の実施例を、図面により詳細に説明する。第5図および第6図は、それぞれ本発明の一実施例を示すメモリへのデータ書き込みの形式図である。第5図および第6図には、メモリ・アドレスが2バイト(16ビット)からなり、正エリアには1アドレスにつき正規データとして1ワード(4バイト)のデータを書く場合の書き込み形式が示される。図では、例えばA₀₀ A₀₁ A₀₂の場合、これらの各文字1つを1バイトとし、アドレスを3バイトで書き込むことを表わしている。25, 27, 29, 31が正エリアであり、26, 28, 30, 32が副エリアである。

第5図(a)では、正エリアのアドレスA_{n0}

A_{n1}に正規データD_{n0} ~ D_{n3}(4バイト)を書き込み、副エリアのアドレスB_{n0} B_{n1}には、チェックデータとしてC_{n0} ~ C_{n5}(6バイト)を書き込むことを示している。

チェックデータは、下記の関係式に基づいてプログラムにより生成される。

$$C_{n0} = \overline{D_{n0}} \dots \dots \dots \quad (1)$$

$$C_{n1} = \overline{D_{n1}} \dots \dots \dots \quad (2)$$

$$C_{n2} = \overline{D_{n2}} \dots \dots \dots \quad (3)$$

$$C_{n3} = \overline{D_{n3}} \dots \dots \dots \quad (4)$$

$$C_{n4} = \overline{D_{n4}} \dots \dots \dots \quad (5)$$

$$C_{n5} = \overline{D_{n5}} \dots \dots \dots \quad (6)$$

ここで、C_{n0}はD_{n0}を反転した値であることを示す。例えば、16進表示で"00"を反転した値は、"FF"である。また、チェックデータを正規データの反転値として書き込む理由は、メモリ素子の故障または周辺回路の故障により内容の如何にかかわらず必ず"1"の出力値となる場合、あるいは必ず"0"の出力値となる場合にも、エラーを検出できるようにするためである。

第5図(a)において、メモリに書き込まれた正規データを $D_{n0w} \sim D_{n3w}$ と記載し、单なる正規データ $D_{n0} \sim D_{n3}$ と区別している。また、チェックデータも $C_{n0w} \sim C_{n3w}$ と記載して、実際にメモリに書き込まれた値を单なるチェックデータ $C_{n0} \sim C_{n3}$ と区別している。データエラーなしに正常に書き込みを行った場合には、当然下記の関係が成立する。

$$D_{n0w} = D_{n0} \dots \dots \dots \quad (7)$$

$$D_{n1w} = D_{n1} \dots \dots \dots \quad (8)$$

$$D_{n2w} = D_{n2} \dots \dots \dots \quad (9)$$

$$D_{n3w} = D_{n3} \dots \dots \dots \quad (10)$$

$$C_{n0w} = C_{n0} \dots \dots \dots \quad (11)$$

$$C_{n1w} = C_{n1} \dots \dots \dots \quad (12)$$

$$C_{n2w} = C_{n2} \dots \dots \dots \quad (13)$$

$$C_{n3w} = C_{n3} \dots \dots \dots \quad (14)$$

$$C_{n4w} = C_{n4} \dots \dots \dots \quad (15)$$

$$C_{n5w} = C_{n5} \dots \dots \dots \quad (16)$$

次に、第5図(b)は、正エリアのアドレス $A_{n0} A_{n1}$ に正規データ $D_{n0} \sim D_{n3}$ (4バ

イト) を書き込み、副エリアのアドレス $B_{n0} B_{n1}$ にはチェックデータとして $C_{n0} \sim C_{n3}$ (4バイト) を書き込む場合を示している。

チェックデータは、下記の関係式に基づいてプログラムにより生成される。

$$C_{n0} = \overline{D_{n0} \oplus D_{n1}} \dots \dots \dots \quad (17)$$

$$C_{n1} = \overline{D_{n2} \oplus D_{n3}} \dots \dots \dots \quad (18)$$

$$C_{n2} = \overline{A_{n0}} \dots \dots \dots \dots \dots \quad (19)$$

$$C_{n3} = \overline{A_{n1}} \dots \dots \dots \dots \dots \quad (20)$$

ここで、 \oplus 記号は Exclusive OR演算を示す。また、 C_{n0} は、 D_{n0} と D_{n1} とで Exclusive OR演算した結果を、反転した値である。 C_{n1} についても、同じである。なお、書き込まれた正規データを $D_{n0} \sim D_{n3}$ の代りに $D_{n0w} \sim D_{n3w}$ と記し、チェックデータを $C_{n0} \sim C_{n3}$ の代りに $C_{n0w} \sim C_{n3w}$ と記載しているのは、第5図(a)の場合と同じである。

第6図(a)は、正エリアのアドレス $A_{n0} A_{n1}$ に正規データ $D_{n0} \sim D_{n3}$ (4バイト) を書き込み、副エリアのアドレス $B_{n0} B_{n1}$ には、チ

ックデータとして C_{n0} と C_{n1} (2バイト) を書き込む場合を示している。

チェックデータは、下記の関係式に基づいて、プログラムにより生成される。

$$C_{n0} = \overline{D_{n0} \oplus D_{n1} \oplus D_{n2} \oplus D_{n3}} \dots \dots \dots \quad (21)$$

$$C_{n1} = \overline{A_{n0} \oplus A_{n1}} \dots \dots \dots \quad (22)$$

ここで、 C_{n0} は $D_{n0} \sim D_{n3}$ の全てを Exclusive ORした結果を反転した値である。

第6図(a)においても、書き込まれた正規データを $D_{n0} \sim D_{n3}$ の代りに $D_{n0w} \sim D_{n3w}$ と記載し、チェックデータを C_{n0} 、 C_{n1} の代りに C_{n0w} 、 C_{n1w} と記載している。

第6図(b)は、正エリアのアドレス $A_{n0} A_{n1}$ に正規データ $D_{n0} \sim D_{n3}$ (4バイト) を書き込み、副エリアのアドレス $B_{n0} B_{n1}$ には、チェックデータとして C_{n0} (1バイト) を書き込む場合を示している。

チェックデータは、下記の関係式に基づき、プログラムにより生成される。

$$C_{n0} = \overline{(D_{n0} \oplus D_{n1} \oplus D_{n2} \oplus D_{n3})} \oplus \overline{A_{n0} \oplus A_{n1}} \dots \dots \dots \quad (23)$$

ここで、 C_{n0} は $D_{n0} \sim D_{n3}$ の全てを Exclusive ORした値である。

第6図(b)で、書き込まれた正規データを $D_{n0w} \sim D_{n3w}$ と記載しているのは、第5図(a)の場合と同じ理由である。

第5図(b)は第5図(a)よりも少ないチェックデータですむようにしており、また第6図(a) (b)では、副エリアに書き込むチェックデータ数をさらに少なくしている。第5図、第6図では、チェックデータを少なくするために、正規データ、または正エリアのアドレス値を Exclusive OR演算した値を用いているが、本発明においては、演算の種類は特定しない。従って、例えば、Exclusive OR演算の代りに和演算でもよく、この場合、演算した結果、桁あふれがあったならば、桁あふれをしたビットを無視して下位8ビットをチェック・バイトとして採用すればよい。

また、前述したように、チェックデータとして、

各正規データバイトを反転した値、または正規データバイト相互の演算結果を反転した値を用いるのは、もし反転しないまま用いると、メモリへの書き込みデータを出力するレジスタ、またはメモリからの読み出しデータが入力されるレジスタの固定的な故障（例えば、レジスタにどのような値をセットしても、レジスタの出力値が常に“00”となるような故障）時に、データエラーを見逃してしまうことがあるからである。

しかし、チェックデータとして、上述のように正規データに対して反転させた値を用いるか、または反転させない値を用いるかは、本発明の本質には関係ないため、演算は特定期にすることにする。

第7図は、第5図(a)に示す形式でメモリ書き込み／読み出しを行うための回路図である。

第7図で、33～38は書き込みデータレジスタ、39, 40はアドレスレジスタ、41はメモリ、42～47は読み出しデータレジスタ、48～51は汎用レジスタである。

第8図は、第7図におけるデータ書き込み時の

アドレス／データエラーの有無の検査のための動作フローチャート、第9図は第7図におけるデータ読み出し時のアドレス／データエラーの有無の検査のための動作フローチャートである。以下、第5図～第9図により、(a)メモリへのデータ書き込み、(b)メモリからのデータの読み出し、(c)アドレス／データエラーの有無のチェック、の順序で、詳述する。

(i) メモリへのデータの書き込み

(イ) プログラムが正エリアのアドレス値 $A_{n_0} \sim A_{n_1}$ をアドレスレジスタ 39 と 40 にセットする命令を実行すると(ステップ 101)、ハードウェアはアドレス値 A_{n_0} をアドレスレジスタ 39 に、 A_{n_1} をアドレスレジスタ 40 に、それぞれ入力する。

(ロ) プログラムが書き込み用レジスタ 33～36 に正規データ $D_{n_0} \sim D_{n_3}$ をセットする命令を実行すると(ステップ 102)、ハードウェアはデータ D_{n_0} をレジスタ 33 に、 \dots データ D_{n_3} をレジスタ 36 に、それぞれ入力する。

それぞれ入力する。

(ト) プログラムが書き込み命令を実行すると(ステップ 103)、ハードウェアはアドレス $A_{n_0} \sim A_{n_1}$ にデータ $D_{n_0} \sim D_{n_3}$ を書き込む。ただし、アドレス $A_{n_0} \sim A_{n_1}$ に実際に書き込まれたデータは $D_{n_0w} \sim D_{n_3w}$ である。データにエラーがなければ、当然(7)～(10)式が成立する。

(ハ) プログラムが書き込み命令を実行すると(ステップ 104)、ハードウェアはアドレス $A_{n_0} \sim A_{n_1}$ にデータ $D_{n_0} \sim D_{n_3}$ を書き込む。ただし、アドレス $A_{n_0} \sim A_{n_1}$ に実際に書き込まれたデータは $D_{n_0w} \sim D_{n_3w}$ である。データにエラーがなければ、当然(7)～(10)式が成立する。

(ニ) プログラムがチェックデータを(1)～(6)式により求める(ステップ 105)。

(ホ) プログラムが副エリアのアドレス値 $B_{n_0} \sim B_{n_1}$ をアドレスレジスタ 39 と 40 にセットする命令を実行すると(ステップ 106)、ハードウェアはアドレス値 B_{n_0} をアドレスレジスタ 39 に、 B_{n_1} をアドレスレジスタ 40 に、それぞれ入力する。

(ヘ) プログラムが書き込み用レジスタ 33～38 にチェックデータ $C_{n_0} \sim C_{n_5}$ をセットする命令を実行すると(ステップ 107)、ハードウェアはチェックデータ C_{n_0} をレジスタ 33 に、 \dots チェックデータ C_{n_5} をレジスタ 38 に、

(ii) メモリからのデータ読み出し(第9図参照)

(イ) プログラムが正エリアのアドレス値 $A_{n_0} \sim A_{n_1}$ をアドレスレジスタ 39 と 40 にセットする命令を実行すると(ステップ 111)、ハードウェアは、アドレス値 A_{n_0} をレジスタ 39 に、アドレス値 A_{n_1} をレジスタ 40 に、それぞれ入力する。

(ロ) プログラムが読み出し命令を実行すると(ステップ 112)、ハードウェアは、アドレス $A_{n_0} \sim A_{n_1}$ から正規データを読み出す。読み出されたデータは、読み出し用レジスタ 42～45 に入力される。このとき、レジスタ 42 の出力値

は $D_{n0r} \dots \dots \dots$ レジスタ 45 の出力値は D_{n3r} となる。読み出し動作中にエラーが発生しなければ、当然。

$$D_{n0r} = D_{n0w} \dots \dots \dots \quad (24)$$

$$D_{n1r} = D_{n1w} \dots \dots \dots \quad (25)$$

$$D_{n2r} = D_{n2w} \dots \dots \dots \quad (26)$$

$$D_{n3r} = D_{n3w} \dots \dots \dots \quad (27)$$

が成立する。

(ハ) プログラムが読み出しデータを汎用レジスタ 48～51 に移す命令を実行すると (ステップ 113)、ハードウェアは、レジスタ 42～45 に格納されたデータ $D_{n0r} \sim D_{n3r}$ を汎用レジスタ 48～51 に入力する。

(ニ) プログラムが副エリアのアドレス値 B_{n0}, B_{n1} をアドレスレジスタ 39 と 40 にセットする命令を実行すると (ステップ 114)、ハードウェアはアドレス値 B_{n0} をレジスタ 39 に、 B_{n1} をレジスタ 40 に、それぞれ入力する。

(ホ) プログラムが読み出し命令を実行すると (ステップ 115)、ハードウェアはアドレス

る。

$$C_{n0r} = \overline{D_{n0r}} \dots \dots \dots \quad (34)$$

$$C_{n1r} = \overline{D_{n1r}} \dots \dots \dots \quad (35)$$

$$C_{n2r} = \overline{D_{n2r}} \dots \dots \dots \quad (36)$$

$$C_{n3r} = \overline{D_{n3r}} \dots \dots \dots \quad (37)$$

$$C_{n4r} = \overline{D_{n4r}} \dots \dots \dots \quad (38)$$

$$C_{n5r} = \overline{D_{n5r}} \dots \dots \dots \quad (39)$$

そして、上式 (34)～(39) の全ての等式が成立したとき、アドレス/データエラーなしと判断するが (ステップ 116～120, 123)、(34)～(39) 式のうちいずれか 1 つでも不成立であれば、アドレス/データエラーありと判断する (ステップ 116～121, 122)。

第 7 図において、正規データとチェックデータの書き込み時と読み出し時を通じて、

- (A) アドレスレジスタ 39, 40
- (B) 書き込み用レジスタ 33～38
- (C) 正エリア
- (D) 副エリア
- (E) 読み出し用レジスタ 42～47

$B_{n0} B_{n1}$ からチェックデータを読み出す。読み出されたデータは、読み出し用レジスタ 42～47 に入力される。このとき、

レジスタ (RDR0) 42 の出力値は C_{n0r}

レジスタ (RDR5) 47 の出力値は C_{n5r} となる。読み出し動作中にデータエラーが発生しなければ、当然、次の式が成立する。

$$C_{n0r} = C_{n0w} \dots \dots \dots \quad (28)$$

$$C_{n1r} = C_{n1w} \dots \dots \dots \quad (29)$$

$$C_{n2r} = C_{n2w} \dots \dots \dots \quad (30)$$

$$C_{n3r} = C_{n3w} \dots \dots \dots \quad (31)$$

$$C_{n4r} = C_{n4w} \dots \dots \dots \quad (32)$$

$$C_{n5r} = C_{n5w} \dots \dots \dots \quad (33)$$

(ホ) プログラムが汎用レジスタ 48～51 にある正規データ $D_{n0r} \sim D_{n3r}$ と、読み出し用レジスタ 42～47 にあるチェックデータ $C_{n0r} \sim C_{n5r}$ とから、下式が成立するか否かを調べ

(F) 汎用レジスタ 48～51

のいずれか 1 箇所で、しかも 1 回だけ故障が生じた場合であれば、第 5 図および第 6 図のどの書き込み形式でメモリに書き込まれていても、本実施によるプログラムのアドレス/データエラー検出動作によりエラーを検出することができる。

しかし、上記各レジスタと各メモリエリアのうち、複数箇所に故障があるとき、または 1 箇所の故障であっても、その故障が正規データとチェックデータの書き込み時と読み出し時を通じて複数回発生すると、故障の状態によっては、第 5 図、第 6 図の書き込み形式ではエラーを見逃してしまうこともあり得る。

このエラーの見逃し率は、チェックデータ数の多い場合ほど小さいので、第 5 図(a)が最も小さく、第 5 図(b)、第 6 図(a)、第 6 図(b)の順に見逃し率は大きくなっていく。

第 1 図は、本発明の一実施例を示すディスク制御系のブロック図である。ここでは、本発明のメモリ書き込み/読み出し方式をディスク制御装置

3内の記憶回路5のメモリに適用した場合を示している。

ディスク制御装置3は、コンピュータシステムにおいて、ディスク装置を制御するものである。また、ディスク制御装置3は、上位装置としてのチャネル2に接続され、下位装置としてのディスク装置にも接続される。ただし、ディスク装置は図示されていない。また、チャネル2はCPU1と接続されている。

ディスク上へ書き込むべきデータは、チャネル2→ディスク制御装置3→ディスク装置の経路で転送され、ディスク上から読み出されたデータは、ディスク装置→ディスク制御装置3→チャネル2の経路で転送される。

これに対して、ディスク制御装置3内の記憶回路5へは、下位に接続されているディスクの管理情報が本実施例の書き込み方式により記憶される。

ディスク管理情報の書き込みは、CPU1からの指示により実行される。チャネル2からは、書き込み管理情報送出信号とともに、書き込み管理

次に、第5図(a)に示す書き込み形式と、第8図、第9図に示すプログラム動作によるアドレス/データエラー検出能力について、詳述する。

(A)書き込み用レジスタ33~36、メモリ41、読み出し用レジスタ42~45、または汎用レジスタ48~51の故障により、データエラーが書き込み時かあるいは読み出し時に発生し、このエラーをプログラムが見逃してしまう場合を考える。このような場合は、読み出した正規データとチェックデータが

$$D_{n+r} \neq D_{n+i} \dots \dots \dots \quad (40)$$

かつ

$$C_{n+r} \neq C_{n+i} \dots \dots \dots \quad (41)$$

であるにもかかわらず、

$$C_{n+r} = \overline{D_{n+r}} \dots \dots \dots \quad (42)$$

が成立してしまう場合である。ここで、iは、
0 ≤ i ≤ 3の範囲の0を含む整数である。

第7図において、正規データ書き込み→正規データ読み出し経路におけるエラー発生確率をPdとし、チェックデータ書き込み経路におけるエラ

情報がディスク制御装置3に送出される。

ディスク制御装置3で受け取られた管理情報は、ディスク制御装置3内のプロセッサ4に入力される。プロセッサ4では、第8図および第9図に示すプログラム動作により、正規データ、つまり管理情報と正規データ書き込みアドレス値からチェックデータを生成する。

正規データとチェックデータは、プロセッサ4から記憶回路5に送られ、正エリアのアドレスに正規データが、また副エリアのアドレスにチェックデータが、それぞれ書き込まれる。

ディスク管理情報の読み出しも、CPU1の指示により行われる。このとき、第8図および第9図に示すプログラム動作により、記憶回路5から正規データとチェックデータが読み出され、アドレス/データエラーの有無が調べられ、エラーがなかったならば、ディスク制御装置3は読み出し管理情報送出信号とともに読み出し管理情報をチャネル2に送出する。もし、エラーがあったときには、エラー報告をチャネル2に対して行う。

一発生確率をPcとし、(42)式が成立する確率をPcd(=1/2⁸)とすると、エラーの見逃し確率Pmisdata₀は、次式で表わされる。

$$P_{misdata_0} = P_d \cdot P_c \cdot P_{cd} \\ = P_d \cdot P_c / 256 \dots \dots \quad (43)$$

一方、これに対して、第4図に示した従来の2重化回路で、正記憶回路のデータエラー経路にエラーが発生する確率をPmとし、副記憶回路のデータ経路にエラーが発生する確率をPsとし、正副両記憶回路のエラーにより誤ったデータ相互が一致する確率をPms(=1/2⁸)とすると、エラーの見逃し確率Pmisdata₁は、次のようになる。

$$P_{misdata_1} = P_m \cdot P_s \cdot P_{ms} \\ = P_m \cdot P_s / 256 \dots \dots \quad (44)$$

ここで、故障について、下記の2種類を定義する。

(a) インタミッテント故障・・・何回かに1回の割合で生じる間欠的な故障

(b) ソリッド故障.....常時の故障

データエラーがインタミッテント故障により生じたものであれば、第4図と第7図のハードウェア量から見て、

$$P_d = P_c = P_m = P_s$$

と考えることができる。ただし、ソリッド故障に対しては、第7図のレジスタ33～38、レジスタ42～47は正規データとチェックデータの書き込み／読み出し時に、いずれのデータに対しても使用されるため、 $P_c = 1$ として考えなければならない。

従って、データエラーに関しては、本発明によるメモリ書き込み／読み出し方式は、従来の2重化記憶回路でのエラー検出能力に比べて、ソリッド故障に対しては少し劣るが、インタミッテント故障に対しては同じであると言える。

(B) アドレスレジスタ39、40またはメモリ内のアドレスラインの故障により、本来、正エリアアドレス $A_{n0} A_{n1}$ と副エリアアドレス $B_{n0} B_{n1}$ に対して、書き込み／読み出しを行うべきところ、不当なアドレスに対して行われ、しか

$$\dots \dots \dots \dots \quad (47)$$

一方、これに対して、第4図に示す2重化回路でアドレスエラーを見逃してしまうのは、正記憶回路のアドレス経路と副記憶回路のアドレス経路の両方に故障があり、正記憶回路と副記憶回路の誤ったアドレス相互が一致してしまう場合である。

アドレスエラーを見逃してしまう確率 $P_{misaddress}$ は、正記憶回路でのアドレスエラーの発生する確率を P_{a0} 、副記憶回路でのアドレスエラーの発生する確率を P_{a1} 、正記憶回路と副記憶回路の誤ったアドレス相互が一致する確率を $P_{a equal} (= 1/2^{16})$ （アドレスが2バイトのとき）とすると、

$$P_{misaddress} = P_{a0} \cdot P_{a1} \cdot P_{a equal} \\ = \frac{P_{a0} \cdot P_{a1}}{2^{16}} \dots \dots \quad (48)$$

第4図において、アドレスは2バイトからなるものとすると、次式の関係があると考えられる。

$$P_{a0} = P_{a0} = P_{a1} \dots \dots \dots \quad (49)$$

これに対して、チェックバイト $C_{m4} r$ と $C_{m5} r$ は、各々1バイトであるため、ハードウェア

もこのアドレスエラーをプログラムが見逃してしまうのは、アドレスエラーのために不当な副エリアアドレス $B_{m0} B_{m1}$ から読み出されたチェックデータ $C_{m4} r$ と $C_{m5} r$ に、さらにデータエラーが加わって、

$$C_{m4} r = A_{n0} \dots \dots \dots \quad (45)$$

かつ

$$C_{m5} r = A_{n1} \dots \dots \dots \quad (46)$$

となってしまった場合である。

アドレス経路でエラーの発生する確率を P_{a0} 、チェックバイト C_{m4} の書き込み／読み出し時にデータエラーの発生する確率を P_{m4} 、チェックバイト C_{m5} の書き込み／読み出し時にデータエラーの発生する確率を P_{m5} 、(45)式が成立する確率を $P_{m4n} (= 1/2^8)$ 、(46)式が成立する確率を $P_{m5n} (= 1/2^8)$ とすると、プログラムがエラーを見逃してしまう確率 $P_{misaddress}$ は、次の式で表わされる。

$$P_{misaddress} = P_{a0} \cdot P_{m4} \cdot P_{m5} \cdot \\ P_{m4n} \cdot P_{m5n} = \frac{P_{a0} \cdot P_{m4} \cdot P_{m5}}{2^{16}}$$

の物量の関係より、次式が成立すると考えられる。

$$P_{m4} r = P_{m5} r < P_{a0} \dots \dots \quad (50)$$

上記(47)～(50)式より、

$$P_{misaddress} < P_{misaddress} \dots \dots \quad (51)$$

となる。

従って、アドレスエラーに関しては、本実施例によるメモリ書き込み／読み出し方式は、従来の2重化回路でのエラー検出能力に比べて、優れていると言える。

このように、本発明のメモリ書き込み／読み出し方式においては、第4図の2重化回路に比べて、少ないハードウェア物量で同程度のアドレス／データエラー検出能力を持たせることができる。特に、正規データを記憶するために必要とする容量が α であり、この α の容量を持つ記憶回路を実現する場合に、適切な容量のメモリ素子が市販されていないという理由で、

$$(正規データ用必要容量) \times 2 \leq (\text{記憶回路容量})$$

である場合には、 α 以上の容量が不要エリアとなる。本発明は、この記憶回路に適用すると、不要

エリアを副エリアとして用いることができるため、ハードウェア物量の増加を伴うことなく、本発明を適用することができる。

〔発明の効果〕

以上説明したように、本発明によれば、必要最小限のハードウェア物量で、従来の2重化回路と同じ程度のエラー検出能力を実現することができる利点がある。

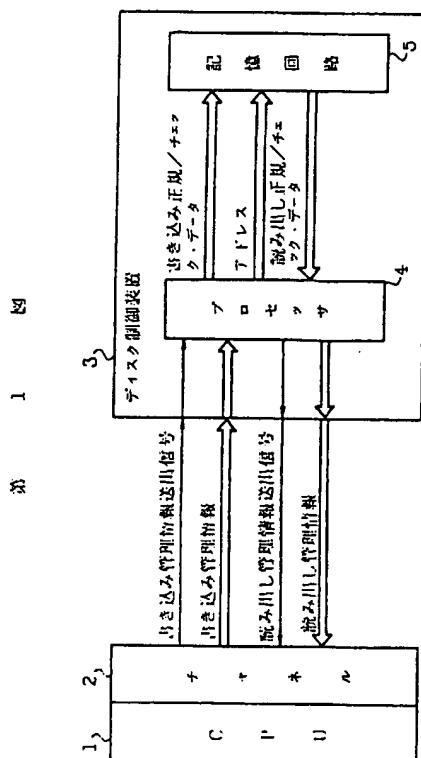
4. 図面の簡単な説明

第1図は本発明の一実施例を示すディスク制御装置の制御系のブロック図、第2図は従来の記憶回路の構成図、第3図は第2図の記憶回路に書き込まれるデータを示す図、第4図は従来の2重化回路の構成図、第5図、第6図は本発明によるアドレス/データエラー検出のための書き込み形式を示す図、第7図は第5図の書き込み形式を行う記憶回路の構成図、第8図、第9図は本発明によるプログラム動作フローチャート。

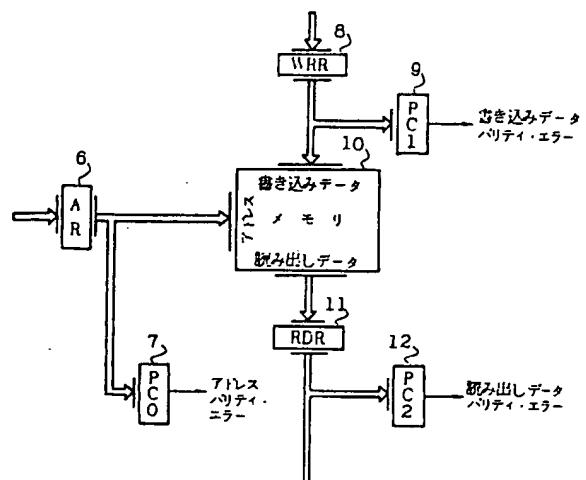
1: C P U 、 2: チャネル、 3: ディスク制御装置、 4: プロセッサ、 5: 記憶回路、 6: アド

レスレジスタ、7：アドレスバリティチェック回路、8, 15, 22, 33～38：書き込み用レジスタ、9：書き込みデータバリティチェック回路、10, 16, 23, 41：メモリ、11, 17, 24, 42～47：読み出し用レジスタ、12：読み出しデータバリティチェック回路、13：メモリエリア、14, 20, 39, 40：アドレスレジスタ、18：書き込みデータコンペアチェック回路、19：アドレスコンペアチェック回路、21：読み出しデータコンペアチェック回路、25, 29, 31：メモリ内正エリア、26, 28, 30, 32：メモリ内副エリア、48～51：汎用レジスタ。

特許出願人 株式会社 日立製作所
代理 人 弁理士 破村 雅俊



第二圖



第3図

(a)

アドレス	A_0	A_1	...	A_n	...			
データ	D_0	P_0	D_1	P_1	...	D_n	P_n	...

13

25

(b)

アドレス	A_0	A_1	...	A_n	...
データ	$D_{0,0}$	$D_{0,1}$...	$D_{n,0}$	$D_{n,1}$
アドレス	$A_{0,0}$	$A_{0,1}$...	$A_{n,0}$	$A_{n,1}$
データ	$C_{0,0}$	$C_{0,1}$...	$C_{n,0}$	$C_{n,1}$

27

(c)

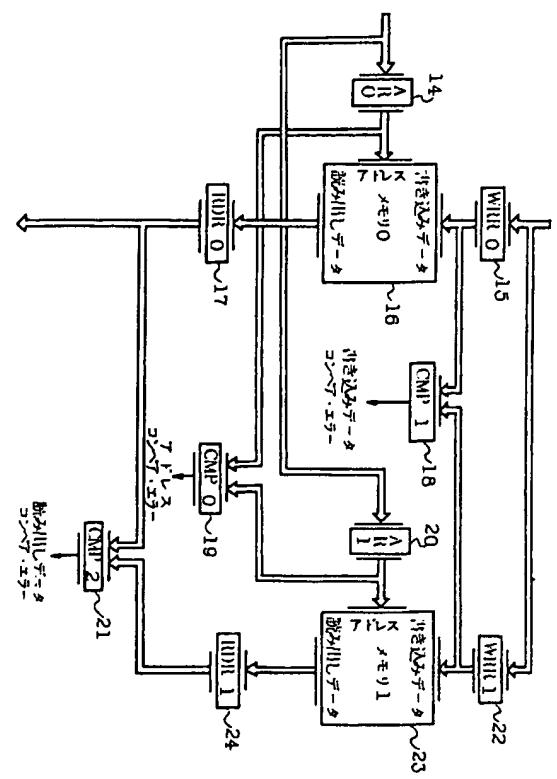
アドレス	A_0	A_1	...	A_n	...
データ	$D_{0,0,0}$	$D_{0,0,1}$...	$D_{n,0,0}$	$D_{n,0,1}$
アドレス	$A_{0,0,0}$	$A_{0,0,1}$...	$A_{n,0,0}$	$A_{n,0,1}$
データ	$C_{0,0,0}$	$C_{0,0,1}$...	$C_{n,0,0}$	$C_{n,0,1}$

(d)

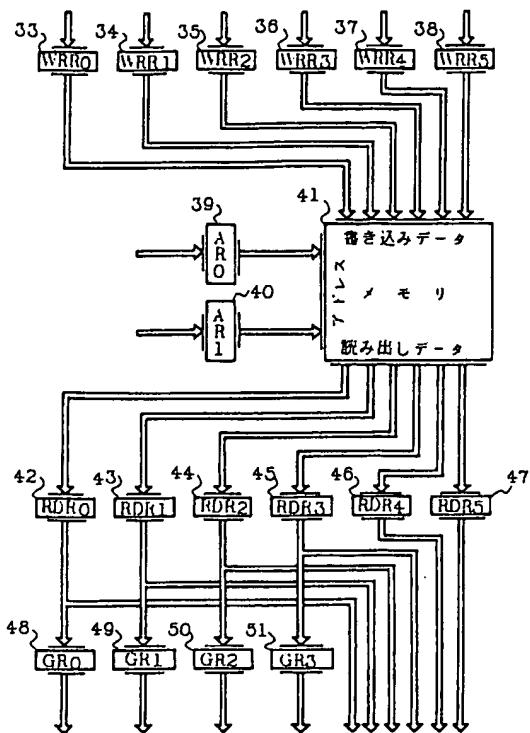
アドレス	A_0	A_1	...	A_n	...
データ	$D_{0,0,0,0}$	$D_{0,0,0,1}$...	$D_{n,0,0,0}$	$D_{n,0,0,1}$
アドレス	$A_{0,0,0,0}$	$A_{0,0,0,1}$...	$A_{n,0,0,0}$	$A_{n,0,0,1}$
データ	$C_{0,0,0,0}$	$C_{0,0,0,1}$...	$C_{n,0,0,0}$	$C_{n,0,0,1}$

(e)

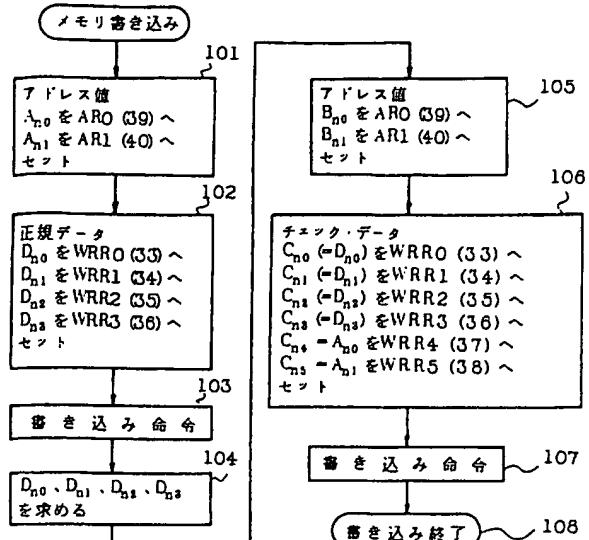
アドレス	A_0	A_1	...	A_n	...
データ	$D_{0,0,0,0,0}$	$D_{0,0,0,0,1}$...	$D_{n,0,0,0,0}$	$D_{n,0,0,0,1}$
アドレス	$A_{0,0,0,0,0}$	$A_{0,0,0,0,1}$...	$A_{n,0,0,0,0}$	$A_{n,0,0,0,1}$
データ	$C_{0,0,0,0,0}$	$C_{0,0,0,0,1}$...	$C_{n,0,0,0,0}$	$C_{n,0,0,0,1}$



第 7 図



第 8 図



第 9 図

